(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **09319635 A**

(43) Date of publication of application: **12.12.97**

(51) Int. Cl **G06F 12/00**
G06F 17/30

(21) Application number: **08136233**

(22) Date of filing: **30.05.96**

(71) Applicant: **MATSUSHITA GRAPHIC COMMUN SYST INC**

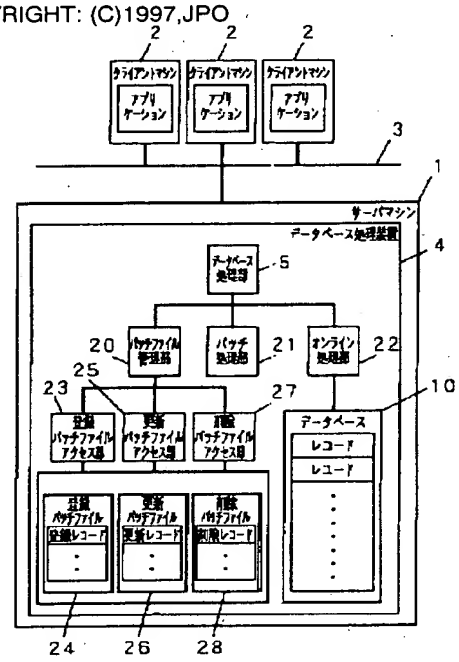
(72) Inventor: **WATANABE SHINICHI**

(54) DATABASE UPDATING METHOD

(57) Abstract:

PROBLEM TO BE SOLVED: To accelerate batch processing or update processing itself of records in a database by performing margin processing to the received update records having the same ID, and updating the records registered in the database while using the records after the margin processing.

SOLUTION: A batch file managing part 20 is activated and it is checked whether the record of the same ID as a record becoming the object of update processing is defined as the object of new registration or not (exists in an update file 26 or not). When the record of the same ID exists in the update batch file 26, the batch file managing part 20 performs the marging processing for the unit of a field to the update record and the update record existent in the update batch file 26 through an update batch file access part 25, generates the new update record, deletes the existent update record and replaces the generated new update record with the deleted update record.



(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平9－319635

(43)公開日　平成９年(1997)12月12日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ			技術表示箇所
Ｇ０６Ｆ	12/00	５１８		Ｇ０６Ｆ	12/00	５１８Ｍ	
	17/30				15/401	３４０Ａ	

審査請求　未請求　請求項の数10　ＯＬ　（全 15 頁）

(21)出願番号　特願平8－136233

(22)出願日　平成８年(1996)５月30日

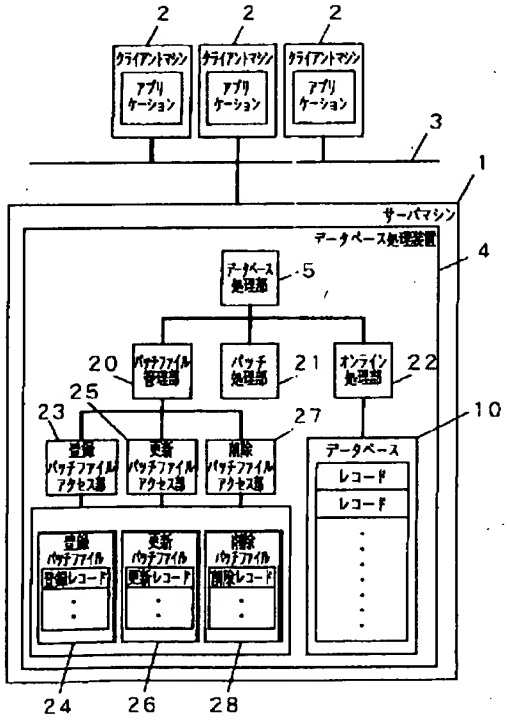
(71)出願人　000187736
松下電送株式会社
東京都目黒区下目黒２丁目３番８号

(72)発明者　渡辺　紳一
東京都目黒区下目黒２丁目３番８号　松下電送株式会社内

(74)代理人　弁理士　滝本　智之　（外１名）

(54)【発明の名称】　データベース更新方法

(57)【要約】
【課題】　バッチ処理の対象となるレコード数の削減、バッチ処理の対象となるレコードデータ量の削減を図ることにより、データベースのレコードの更新を迅速に行なうこと。
【解決手段】　データベースに記録されたレコードに対する更新指示と更新レコードとを順次受付け、同一ＩＤを有する既登録の更新レコードに対してマージ処理を行ない、データベースのレコードをそのマージ処理後のレコードを用いて更新する構成とした。



1

【特許請求の範囲】

【請求項１】　データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、同一ＩＤを有する受付済の更新レコードに対してマージ処理を行ない、データベースのレコードをそのマージ処理後のレコードを用いて更新することを特徴とするデータベース更新方法。

【請求項２】　データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、その更新レコードと同一ＩＤを有する受付済の更新レコードとをマージ処理して生成した単一の新更新レコードを更新レコードとして保持した後、データベースのレコードをこの新更新レコードにより更新することを特徴とするデータベース更新方法。

【請求項３】　データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、順次受付けた同一ＩＤを有する複数の更新レコードをマージ処理してＩＤ毎に単一のファイルとした上で、前記データベースのレコードの更新処理を実行することを特徴とするデータベース更新方法。

【請求項４】　マージ処理により生成される新更新レコードを順次保存する一方、マージ処理の対象としたその更新レコードと同一ＩＤを有する受付済の更新レコードを順次消去することを特徴とする請求項１乃至請求項３記載のデータベース更新方法。

【請求項５】　最終的に削除対象となるレコードと同一レコードに対する登録指示又は更新指示がある場合には、その登録指示又は更新指示を無効にすることを特徴とする請求項１乃至請求項４記載のデータベース更新方法。

【請求項６】　データベースのレコードの更新処理実行前の更新レコードを登録処理バッチファイルと更新処理バッチファイルと削除処理バッチファイルとに分割して管理することを特徴とする請求項１乃至請求項４記載のデータベース更新方法。

【請求項７】　登録バッチファイル又は更新バッチファイルに存在する削除レコードと同一ＩＤのレコードをデータベースのレコード更新以前に消去することを特徴とする請求項６記載のデータベース更新方法。

【請求項８】　レコードの更新指示を時刻管理し、同一レコードに対して複数の更新指示がある場合は、新しい時刻の更新指示のみを実行することを特徴とする請求項１乃至請求項７記載のデータベース更新方法。

【請求項９】　データベースに対する検索の実行の後に、その検索結果レコードと更新処理待ちの更新レコードとをマージ処理することを特徴とする請求項１乃至請求項８記載のデータベース更新方法。

【請求項１０】　データベースに登録されたレコードの更新指示受付けの際にレコードの更新に要する時間を計算し、その時間が一定値以下である場合には、更新処理

2

をリアルタイムで実行し、その時間が一定値以上である場合には、更新処理をバッチ処理で実行することを特徴とする請求項１乃至請求項９記載のデータベース更新方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ネットワークによって接続されたクライアント、サーバ型データベースシステムにおけるレコードの登録、更新、削除等の更新処理をバッチ処理で実行するデータベース更新方法に関する。

【０００２】

【従来の技術】近年、ネットワークの普及に伴いクライアント、サーバ型データベースシステムが増加している。このようなＬＡＮ等のネットワークによって接続されたクライアント、サーバ型データベースシステムにおいて、大量データを処理する場合などレコードに対する登録、更新、削除処理が遅い場合には、検索処理はオンラインでリアルタイムに処理し、登録、更新、削除のデータの更新処理は登録、更新、削除レコードをオンラインでバッチファイルに保持し、オフライン時にバッチ処理で処理する形態が取られている。

【０００３】以下に、このような従来のクライアント、サーバ型データベースシステムについて図面を用いて説明する。図１４は、従来のクライアント、サーバ型データベースシステムを示すものである。

【０００４】図１４において、１は、ＷＳ、ＰＣ、ホストマシン等のサーバマシン、２は、検索、登録、更新、削除処理を要求するアプリケーションを搭載した、ＷＳ、ＰＣ等のクライアントマシン、であり、サーバマシン１とクライアントマシン２とは、ＬＡＮ、ＷＡＮ等のネットワーク３で接続されている。４は、サーバマシン1に実装されたデータベース処理装置、５は、クライアントからの処理要求を受取り、オンライン時にクライアントからの処理要求を受取り、検索、登録、更新、削除処理を実行するデータベース処理部、６は、クライアントマシン2からのレコードの登録、更新、削除の要求に応じ、オンライン時にリアルタイムにバッチファイル７を作成するバッチファイル管理部、８は、オフライン時にバッチファイルのレコードのバッチ処理を実行するバッチ処理部、９は、オンライン時にリアルタイムにデータベース１０を構成するレコードの検索処理等を実行するオンライン処理部である。

【０００５】データベース１０に登録されているレコードは、例えば、図１５に示す形態であり、１レコードは、レコードＩＤ１１と、そのデータ内容が、それぞれ、”ＡＡＡ”、”ＢＢＢ”、”ＣＣＣ”である３つのフィールド１２とから構成されている。また、図１７は、バッチファイル７内の１レコードを示したものであり、レコードＩＤ１１とフィールド１２の他に登録、更

新、削除の処理の別を示す処理種別１３を示すフィールドを有している。このバッチファイル７内に登録されている各レコードは、順次時系列に受付けられる。

【０００６】以上のように構成されたクライアント、サーバ型データベースシステムの動作概要を、図１６のフロー図に沿って説明する。

【０００７】まず、クライアントマシン２のアプリケーションは、サーバマシン１のデータベース処理装置４に対し、レコードの検索処理、あるいは、登録処理、あるいは、更新処理、あるいは、削除処理を依頼し（ＳＴ１）、データベース処理部５が、その要求を受け付け（ＳＴ２）、各処理種別に応じて次の処理を行う。

【０００８】検索処理の場合は、オンライン処理部９を起動し（ＳＴ３）、データベース１０から検索条件に合致したレコードを検索し、検索結果レコードをクライアントマシンへ返す（ＳＴ４）。

【０００９】登録処理、更新処理、削除処理の場合には、バッチファイル管理部６を起動し（ＳＴ５）、オンラインでバッチファイル７へ登録レコード、更新レコード、削除レコードを図１７のように時系列に順次追加し（ＳＴ６）、オフライン時に、バッチ処理部８で、データベース１０へのレコードの登録処理、更新処理、削除処理を実行する（ＳＴ７）。

【００１０】

【発明が解決しようとする課題】しかしながら、上記の従来のクライアント、サーバ型データベースシステムの構成では、時系列にバッチファイルが作成されるため、オフライン時のバッチ処理の際に、同一レコードに対する更新、削除処理がその都度実行されるので、これらの処理が大量に発生する場合には、膨大に時間を要するという問題がある。

【００１１】また、バッチ処理そのものの問題として、バッチファイルへの登録を完了しても、データベースのレコードの実体が更新されなければ、検索対象にはならないため検索結果に反映されないという問題がある。

【００１２】また、レコードのデータ量とサーバマシンの負荷等との関係で、登録、更新、削除等の処理がリアルタイムで実行できる程に短時間で済む場合でも、バッチ処理対象となり、実際の更新処理が遅れ、更新レコードの増大に応じて、オフライン時のバッチ処理の負荷も増大するという問題がある。

【００１３】本発明は、これらの課題に鑑みて為されたものであり、バッチ処理の対象となるレコード数の削減、バッチ処理の対象となるレコードデータ量の削減を図ることにより、データベースのレコードの更新を迅速に行なうとともに、バッチ処理自体の負荷を軽減することにより、バッチ処理、データベースのレコードの更新処理自体の迅速化を実現するデータベース更新方法を提供することを目的とする。

【００１４】

【課題を解決するための手段】これらの目的を達成するために、本発明のデータベース更新方法は、データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、同一ＩＤを有する受付済の更新レコードに対してマージ処理を行ない、データベースのレコードをそのマージ処理後のレコードを用いて更新するという構成をとる。

【００１５】これにより、データベースのレコードの更新を迅速に行うとともに、バッチ処理自体の負荷を軽減することにより、バッチ処理、データベースのレコードの更新処理自体の迅速化を実現することができる。

【００１６】

【発明の実施の形態】本発明の請求項１記載の発明は、データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、同一ＩＤを有する受付済の更新レコードに対してマージ処理を行ない、データベースのレコードをそのマージ処理後のレコードを用いて更新するようにしたものである。

【００１７】また、請求項２記載の発明は、データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、その更新レコードと同一ＩＤを有する受付済の更新レコードとをマージ処理して生成した単一の新更新レコードを更新レコードとして保持した後、データベースのレコードをこの新更新レコードにより更新するようにしたものである。

【００１８】また、請求項３記載の発明は、データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、順次受付けた同一ＩＤを有する複数の更新レコードをマージ処理してＩＤ毎に単一のファイルとした上で、前記データベースのレコードの更新処理を実行するようにしたものである。

【００１９】請求項１乃至請求項３記載の発明により、同一レコードに対する複数の更新処理を単一のアクセスにより一回で処理することができるため、バッチ処理、データベースのレコードの更新処理自体の迅速化を実現できる。

【００２０】請求項４記載の発明は、請求項１乃至請求項３記載のデータベース更新方法において、マージ処理により生成される新更新レコードを順次保存する一方、マージ処理の対象としたその更新レコードと同一ＩＤを有する受付済の更新レコードを順次消去するようにしたものである。これにより、マージ処理実行の度に不要となった旧レコードが消去され、更新レコードを保持するメモリの有効活用ができ、バッチ処理、データベースのレコードの更新処理自体の迅速化も実現できる。

【００２１】請求項５記載の発明は、請求項１乃至請求項４記載のデータベース更新方法において、最終的に削除対象となるレコードと同一レコードに対する登録指示又は更新指示がある場合には、その登録指示又は更新指示を無効にするようにしたものである。これにより、後

に削除されるレコードについての無駄な登録処理や更新処理を実行する必要がなくなるため、バッチ処理、データベースのレコードの更新処理自体の迅速化が実現できる。

【００２２】請求項６記載の発明は、請求項１乃至請求項４記載のデータベース更新方法において、データベースのレコードの更新処理実行前の更新レコードを登録バッチファイルと更新バッチファイルと削除バッチファイルとに分割して管理するようにしたものである。これにより、予め処理ファイルをグルーピングして整列するため、ファイル管理が容易になり、バッチ処理、データベースのレコードの更新処理自体を簡易、迅速に実行できる。

【００２３】請求項７記載の発明は、請求項６記載のデータベース更新方法において、登録バッチファイル又は更新バッチファイルに存在する削除レコードと同一ＩＤのレコードをデータベースのレコード更新以前に消去するようにしたものである。これにより、更新レコードを保持するメモリの有効活用ができ、バッチ処理、データベースのレコードの更新処理自体の迅速化も実現できる。

【００２４】請求項８記載の発明は、請求項１乃至請求項７記載のデータベース更新方法において、レコードの更新指示を時刻管理し、同一レコードに対して複数の更新指示がある場合は、新しい時刻の更新指示のみを実行するようにしたものである。これにより、同一レコードに再度更新がかけられた結果不必要となった旧更新処理をバッチ処理対象から削除できるため、バッチレコード数を削減できバッチ処理の負荷を軽減させることができる。

【００２５】請求項９記載の発明は、請求項１乃至請求項７記載のデータベース更新方法において、データベースに対する検索の実行の後に、その検索結果レコードと更新処理待ちの更新レコードとをマージ処理するようにしたものである。これにより、データベースのレコードの実体に対するバッチ処理に時間がかかり更新が遅れた場合であっても、クライアントが実行する検索の結果は、常に、更新後の最新のものとなる。

【００２６】請求項１０記載の発明は、請求項１乃至請求項７記載のデータベース更新方法において、データベースに登録されたレコードの更新指示受付けの際にレコードの更新に要する時間を計算し、その時間が一定値以下である場合には、更新処理をリアルタイムで実行し、その時間が一定値以上である場合には、更新処理をバッチ処理で実行するようにしたものである。これにより、短時間で更新処理が完了する場合には即時に更新処理を実行し、更新処理に時間がかかる場合には、バッチ処理により更新処理を実行するため、クライアント、サーバ型データベースシステム全体として最適の運用が可能になる。

【００２７】以下、本発明の実施の形態について、図面を用いて、具体的に説明する。図１は、本発明のデータベース更新方法を実現するクライアント、サーバ型データベースシステムの全体構成図である。図１において、１は、ＷＳ、ＰＣ、ホストマシン等のサーバマシン、２は、検索、登録、更新、削除処理を要求するアプリケーションを搭載した、ＷＳ、ＰＣ等のクライアントマシンであり、サーバマシン１とクライアントマシン２とは、ＬＡＮ、ＷＡＮ等のネットワーク３で接続されている。４は、サーバマシン1に実装されたデータベース処理装置、５は、クライアントからの処理要求を受取り、オンライン時にクライアントからの処理要求を受取り、検索、登録、更新、削除処理を実行するデータベース処理部、２０は、クライアントマシン2からのレコードの登録、更新、削除の要求に応じ、オンライン時にリアルタイムにバッチファイルを作成するバッチファイル管理部、２１は、オフライン時にバッチファイルのレコードのバッチ処理を実行するバッチ処理部２２は、オンライン時にリアルタイムにデータベース１０を構成するレコードの検索処理等を実行するオンライン処理部である。また、２３は、登録バッチファイル２４への登録レコードの追加、更新、削除処理を行う登録バッチファイルアクセス部、２５は、更新バッチファイル２６への更新レコードの追加、更新、削除処理を行う更新バッチファイルアクセス部、２７は、削除バッチファイル２８への削除レコードの追加、更新、削除処理を行う削除バッチファイルアクセス部である。

【００２８】以上のように構成されたクライアント、サーバ型データベースシステムデータベース処理装置におけるデータベース更新方法について、具体的に、図面を用いて説明する。図２は、本発明のデータベース更新方法の全体処理フロー図である。

【００２９】まず、クライアントマシン２のアプリケーションは、サーバマシン１のデータベース処理装置４に対し、登録、更新、削除レコードを送信し、登録処理、更新処理、削除処理、検索処理を依頼する（ＳＴ２０１）。データベース処理部５は、登録、更新、削除レコードを受取り、登録処理、更新処理、削除処理、検索処理要求を受け付けて（ＳＴ２０２）、依頼内容に応じて異なる制御を実行する。

【００３０】検索処理の場合には、データベース処理部5は、オンライン処理部２２を起動し（ＳＴ２０３）、検索を実行し、データベースから検索条件に合致したレコードを取得し、クライアントマシン2のアプリケーションへ検索結果を返す（ＳＴ２０４）。

【００３１】登録処理の場合には、データベース処理部5は、バッチファイル管理部２０を起動し（ＳＴ２０５）、バッチファイル管理部２０が、登録バッチファイルアクセス部２３により登録バッチファイル２４に登録レコードを追加する（ＳＴ２０６）。

【００３２】次に、更新処理について説明する。更新処理の場合には、データベース処理部5は、図３の更新処理１の処理フロー図に従って、更新処理を実行する。まず、バッチファイル管理部２０が起動され（ＳＴ３０１）、更新処理の対象となっているレコードと同一ＩＤのレコードが新規登録の対象となっていないかをチェックする。具体的には、登録バッチファイルアクセス部２３により更新レコードと同一ＩＤのレコードが登録バッチファイル２４にあるか否かチェックする（ＳＴ３０２）。

【００３３】登録バッチファイル２４内に同一レコードがある場合には、登録バッチファイルアクセス部２３により、更新レコードと登録バッチファイル２４内の既存の登録レコードとをフィールド単位でマージ処理し新規登録レコードを生成した後、既存の登録レコードを削除し、生成した新規登録レコードを削除した既存の登録レコードと置換し新たに追加する（ＳＴ３０３）。例えば、図４に示すように、レコードＩＤが１００のレコードのフィールド１を”ささささ”に更新する更新レコード２９をクライアントのアプリケーションから受け取った場合、登録バッチファイルに既に登録されている登録レコード３０は、フィールド１が”あああ”から”ささささ”に訂正され新規登録レコード３１として登録バッチファイルに記憶される。

【００３４】登録バッチファイル２４内に同一レコードがない場合には、次に、バッチファイル管理部２０は、更新バッチファイルアクセス部２５により、更新レコードと同一ＩＤのレコードが更新バッチファイル２６にあるか否かをチェックする（ＳＴ３０４）。

【００３５】同一ＩＤのレコードが更新バッチファイル２６にある場合には、バッチファイル管理部２０は、更新バッチファイルアクセス部２５により、更新レコードと更新バッチファイル２６内の既存の更新レコードとをフィールド単位でマージ処理し新規更新レコードを生成した後、既存の更新レコードを削除し、生成した新規更新レコードを削除した更新レコードと置換する（ＳＴ３０５）。例えば、図５に示すように、レコードＩＤが１００のレコードのフィールド１を”ささささ”に更新する更新レコード３２をクライアントのアプリケーションから受け取った場合、更新バッチファイル２６に既に登録されているレコードＩＤが１００のレコードのフィールド３を”すすす”に更新する更新レコード３３は、フィールド１を”ささささ”に、フィールド３を”すすす”に更新する更新レコード３４に更新される。

【００３６】同一ＩＤのレコードが更新バッチファイル２５にない場合には、次に、バッチファイル管理部２０は、削除バッチファイルアクセス部２７により、更新レコードと同一ＩＤのレコードが削除バッチファイル２８にあるか否かをチェックする（ＳＴ３０６）。同一ＩＤのレコードが削除バッチファイル２８にある場合には、

その更新レコードは、更新不要のレコードであるから、バッチファイル管理部２０は、受付けた更新レコードを破棄する（ＳＴ３０７）。

【００３７】同一ＩＤのレコードが削除バッチファイル２８にない場合に、初めて、バッチファイル管理部２０は、更新バッチファイルアクセス部２５により、更新バッチファイル２６へ受付けた更新レコードを追加する（ＳＴ３０８）。

【００３８】このようにして、更新レコードと同一ＩＤのレコードが登録バッチファイル及び削除バッチファイルに存在するかどうかをチェックし、既に登録バッチファイルか更新バッチファイルに同レコードが存在する場合には、双方のレコードをフィールド単位でマージ処理したレコードで既存のバッチファイル中のレコードを更新する処理を、更新レコードを受付ける度に実行することにより、同一のレコードに対する登録処理、更新処理を、常に、単一のレコードを基準として管理し処理できるため、バッチ処理対象となる実レコード数の飛躍的削減が可能となる。また、既に削除バッチファイルに同一ＩＤのレコードが保持されていることが確認できた場合には、同一ＩＤの登録レコード、更新レコード全てが消去されるため、バッチレコード数を大幅な削減が可能となる。

【００３９】次に、削除処理について説明する。削除処理の場合には、データベース処理部は、図６の削除処理１のフロー図に従って、削除処理を実行する。まず、データベース処理部５は、バッチファイル管理部２０を起動する（ＳＴ６０１）。

【００４０】バッチファイル管理部２０は、登録バッチファイルアクセス部２３により削除レコードと同一ＩＤのレコードが登録バッチファイル２４にあるか否かをチェックする（ＳＴ６０２）。ある場合は、バッチファイル管理部２０は、登録バッチファイル２４からそのレコードを削除する（ＳＴ６０３）。

【００４１】無い場合は、次に、バッチファイル管理部２０は、削除レコードと同一ＩＤのレコードが更新バッチファイル２６にあるか否かをチェックする（ＳＴ６０４）。ある場合には、バッチファイル管理部２０は、更新バッチファイル２６からそのレコードを削除する（ＳＴ６０５）。

【００４２】無い場合には、次に、バッチファイル管理部２０は、削除レコードと同じレコードが削除バッチファイル１１９にあるか否かをチェックする（ＳＴ６０６）。ある場合には、その削除レコードは、重複していることになり再登録の必要はないため、バッチファイル管理部２０は、受付けた削除レコードを破棄する（ＳＴ６０７）。

【００４３】同一ＩＤのレコードが削除バッチファイル２８にない場合、バッチファイル管理部２０は、初めて、削除バッチファイル２８に受付けた削除レコードを

追加する（ＳＴ６０８）。

【００４４】このようにして、登録バッチファイル又は更新バッチファイルに存在する削除レコードと同一ＩＤのレコードは、その登録処理、更新処理自体が必要ないわけであるから、これら全てを消去することにより、バッチレコード数が削減されることとなる。例えば、図７のように更新処理を時系列に受付けた場合、従来のデータベース処理装置におけるバッチ処理では合計８レコード全てについての処理が必要となるが、本発明によれば、更新等の処理を受付ける度に、同一ＩＤのレコードの検索、マージ処理の実行をしてバッチファイルの集約処理を行う結果、バッチ処理の対象レコード数は、図８に示すように、合計４レコードになる。

【００４５】以上の処理により、データベース１０に対する登録、更新、削除待ちのバッチファイルは、各レコードＩＤにつき単一のレコードに集約されるため、本発明では、更に、これをクライアントマシン２からの検索依頼があった場合に、その検索結果に反映させることとした。

【００４６】この処理は、オンライン処理部２２の機能の一部として実行される。図９の検索処理フロー図を用いて具体的に説明する。まず、クライアントマシン２アプリケーションから検索処理要求を受け取った場合、データベース処理部５は、オンライン処理部２２を起動し（ＳＴ９０１）、検索を実行し、データベース１０から検索条件に合致したレコードを取得し保持する（ＳＴ９０２）。

【００４７】次いで、オンライン処理部２２は、検索結果のレコードの集合から、1レコードを取り出し（ＳＴ９０４）、バッチファイル管理部２０を起動して（ＳＴ９０５）、各バッチファイルに同一ＩＤのレコードがあるか否かをチェックする。

【００４８】バッチファイル管理部２０は、先ず、削除バッチファイルアクセス部２７により、削除バッチファイル２８に同一ＩＤのレコードがあるかチェックさせる（ＳＴ９０６）。ある場合は、これをオンライン処理部２２に通知し（ＳＴ９０７）、オンライン処理部２２により、検索結果レコードから該当レコードを削除する（ＳＴ９０８）。

【００４９】削除バッチファイル２８に同一ＩＤのレコードが無い場合は、次に、バッチファイル管理部２０は、更新バッチファイルアクセス部２５により、更新バッチファイル２６に同一ＩＤのレコードがあるか否かをチェックする（ＳＴ９０９）。

【００５０】更新バッチファイル２６に同一ＩＤのレコードがある場合は、更新バッチファイルアクセス部２５は、更新バッチファイル２６の更新レコードが検索条件に合致するかチェックし（ＳＴ９１０）、合致する場合には、同一ＩＤのレコードが更新バッチファイル２６に存在することを、オンライン処理部２２に通知し（ＳＴ９１１）、オンライン処理部２２は検索結果レコードと更新バッチファイル２６の更新レコードとのマージ処理を実行し、新たに生成された新規更新レコードで検索結果レコードを更新する（ＳＴ９１２）。合致しない場合には、それをオンライン処理部２２に通知して、検索結果レコードを破棄する（ＳＴ９１３）。これは、バッチ処理実行前であるためデータベース１０の実レコードは更新されていないものの、既にいずれかのクライアント２からのレコード更新要求が受付けられており、現在の検索結果レコードは、本来検索条件には、合致しないレコードであるからである。

【００５１】更新バッチファイル２６に同一ＩＤのレコードが無い場合は、次の検索結果レコードについて同様の処理を実行する（ＳＴ９１４）。

【００５２】このように、ＳＴ９０４からＳＴ９１４までの処理を、データベースから検索した検索結果レコードがなくなるまで繰返して実行する（ＳＴ９０３）。

【００５３】以上の削除バッチファイル２８と更新バッチファイル２６との処理ループを抜けた後、オンライン処理部２２は、登録バッチファイルアクセス部２３により、登録バッチファイル２４の登録レコードに対し検索を実行し、検索条件に合致した登録レコードを取得し（ＳＴ９１５）、オンライン処理部２２は、該当レコードを検索結果レコードに追加する（ＳＴ９１６）。

【００５４】以上の検索処理が実行された後、データベース処理部５は、クライアントのアプリケーションに対して検索結果レコードを送信する。

【００５５】以上の処理により、データベース１０の実レコードの更新前であっても、結果的に検索対象とされるため、レコードの更新にバッチ処理を採用しつつも、クライアントに対しては最新の検索結果を提供することができ、検索のリアルタイム性を確保できる。

【００５６】次に、本発明の第二の実施の態様を説明する。本実施態様では、更に、バッチ処理部２１による、データベース１０に対する各バッチファイルの更新処理の実行タイミングを、システム全体のパフォーマンスを考慮して調整しており、データベース１０の更新処理が短時間で完了する場合はオンラインにより実行し、時間を要する場合はバッチ処理により実行することとしている。図１０および図１１の処理フロー図を用いて具体的に説明する。

【００５７】この処理も、オンライン処理部２２の機能の一部として実行される。まず、クライアントマシン２のアプリケーションは、データベース処理装置４に対し、登録処理、更新処理、削除処理、検索処理を依頼し（ＳＴ１００１）、データベース処理部５は、これを受付け（ＳＴ１００２）、検索処理であればこれを実行する（ＳＴ１００３）。

【００５８】登録処理の場合には、データベース処理部５は、ＣＰＵ負荷状況、データベース内レコード数を取

得し、それらの値より現負荷時の登録処理時間を計算し、規定時間で処理可能かをチェックする（ＳＴ１００４）。その際、例えば、登録処理に対する規定時間、負荷なし時リアルタイム処理時間［対単位レコード長］、処理時間増加比率［対ＣＰＵ負荷］、処理時間増加比率［対単位ＤＢレコード数］はあらかじめ図１２の処理時間テーブルのように設定し、その計算式は図１３のように設定する。

【００５９】規定時間で済む場合には、データベース処理部５は、オンライン処理部２２により、データベース１０に対しリアルタイムに登録処理を実行し（ＳＴ１００５）、規定時間を超える場合には、通常の場合と同様に登録バッチファイルへ登録処理を行う（ＳＴ１００６からＳＴ１００７）。

【００６０】例えば、登録レコード長が２００バイト、ＯＳから取得したサーバマシンのＣＰＵ負荷が３０％、データベースレコード数が１００レコードとすると、現負荷時の１レコードあたりのリアルタイム登録処理時間は、６．８秒となる（図１２の設定で図１３の計算式による）。仮に、あらかじめ設定された登録処理に対する規定時間が７秒だとすると、規定時間で処理可能なため、この登録処理は、データベース１０へリアルタイムに登録される。

【００６１】更新処理の場合には、データベース処理部８１０は、ＣＰＵ負荷状況、データベース内レコード数、を取得しそれらの値より現負荷時の更新処理時間を計算し、現負荷時のリアルタイム更新処理時間の規定時間で処理可能かチェックする（ＳＴ１０１０）。規定時間で済む場合には、データベース処理部５は、バッチファイル管理部２０により、登録、更新、削除バッチファイルに同レコードが存在するかチェックする（ＳＴ１０１１）。存在する場合には、そのままオンラインでデータベース１０のレコードを更新すると各バッチファイルのレコードと不整合が発生するため、既に説明したマージ処理を含む各バッチファイルに対する更新処理を実行する（ＳＴ１０１２）。存在しない場合にはリアルタイムに更新処理を実行する（ＳＴ１０１３）。

【００６２】規定時間を超える場合には、通常の更新と同様に各バッチファイルに対する更新処理を実行する（ＳＴ１０１４）。

【００６３】削除処理の場合にも、同様に、削除処理時間を計算し、規定時間で処理可能かチェックする（ＳＴ１０１５）。規定時間で済む場合は、バッチファイル管理部２０により、登録、更新、削除バッチファイルに同レコードが存在するかチェックし（ＳＴ１０１６）、存在する場合は各バッチファイルに対する削除処理を実行し（ＳＴ１０１７）、存在しない場合はリアルタイムに削除処理を実行する（ＳＴ１０１８）。規定時間を超える場合は、各バッチファイルに対する削除処理を実行する（ＳＴ１０１９）。

【００６４】以上のように、レコードのデータ量やサーバマシンの負荷が少なく、データベースへのリアルタイムでの登録、更新、削除処理が短時間で済む場合、リアルタイムでデータベースノレコードに対する処理を実行することにより、オフライン時のバッチ処理の負荷を減少させることができる。

【００６５】

【発明の効果】本発明により、同一レコードに対する複数の更新処理を単一のアクセスにより一回で処理することができるため、バッチ処理、データベースのレコードの更新処理自体の迅速化を実現できる。

【００６６】また、マージ処理実行の度に不要となった旧レコードが消去され、更新レコードを保持するメモリの有効活用ができ、バッチ処理、データベースのレコードの更新処理自体の迅速化も実現できる。

【００６７】また、後に削除されるレコードについての無駄な登録処理や更新処理を実行する必要がなくなるため、バッチ処理、データベースのレコードの更新処理自体の迅速化が実現できる。

【００６８】また、予め処理ファイルをグルーピングして整列するため、ファイル管理が容易になり、バッチ処理、データベースのレコードの更新処理自体を簡易、迅速に実行できる。

【００６９】また、更新レコードを保持するメモリの有効活用ができ、バッチ処理、データベースのレコードの更新処理自体の迅速化も実現できる。

【００７０】また、同一レコードに再度更新がかけられた結果不必要となった旧更新処理をバッチ処理対象から削除できるため、バッチレコード数を削減できバッチ処理の負荷を軽減させることができる。

【００７１】また、データベースのレコードの実体に対するバッチ処理に時間がかかり更新が遅れた場合であっても、クライアントが実行する検索の結果は、常に、更新後の最新ものとなる。

【００７２】また、短時間で更新処理が完了する場合には即時に更新処理を実行し、更新処理に時間がかかる場合には、バッチ処理により更新処理を実行するため、クライアント、サーバ型データベースシステム全体として最適の運用が可能になる。

【図面の簡単な説明】

【図１】本発明のデータベース更新方法を実現するクライアント、サーバ型データベースシステムの全体構成図

【図２】本発明のデータベース更新方法の全体処理フロー図

【図３】本発明のデータベース更新方法の更新処理フロー図

【図４】本発明の更新バッチファイル内のレコードを表す図

【図５】本発明の更新バッチファイル内のレコードを表す図

13

【図６】本発明のデータベース更新方法の削除処理フロー図

【図７】本発明の時系列更新依頼レコードを表す図

【図８】本発明の各バッチファイル内のレコードを表す図

【図９】本発明のデータベース更新方法の検索処理フロー図

【図１０】本発明の第２実施態様によるデータベース更新方法の処理フロー図

【図１１】本発明の第２実施態様によるデータベース更新方法の処理（続き）フロー図

【図１２】本発明の第２実施態様によるデータベース更新方法に使用する処理時間テーブルを表す図

【図１３】本発明の第２実施態様によるデータベース更新方法に使用する処理時間の計算方法を示す図

14

【図１４】従来のデータベース更新方法を実現するクライアント、サーバ型データベースシステムの全体構成図

【図１５】従来のデータベース内のレコードを表す図

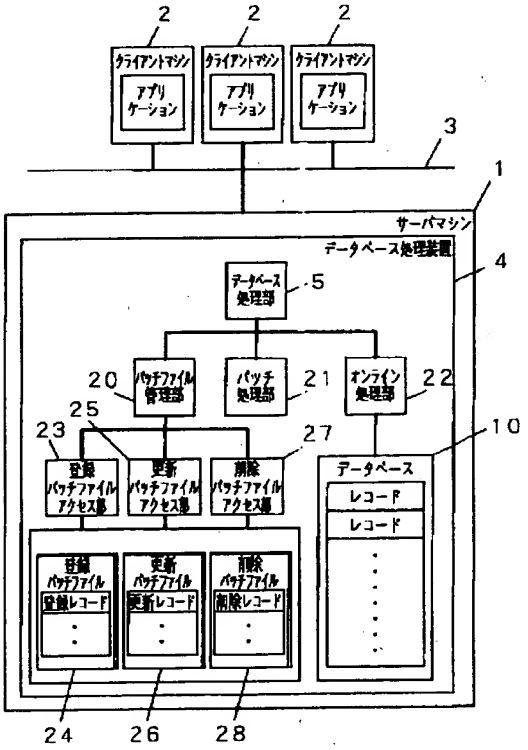
【図１６】従来のデータベース更新方法の全体処理フロー図

【図１７】従来のバッチファイル内のレコードを表す図

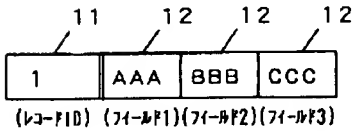
【符号の説明】

１０　データベース
２０　バッチファイル管理部
２１　バッチ処理部
２２　オンライン処理部
２４　登録バッチファイル
２６　更新バッチファイル
２８　削除バッチファイル

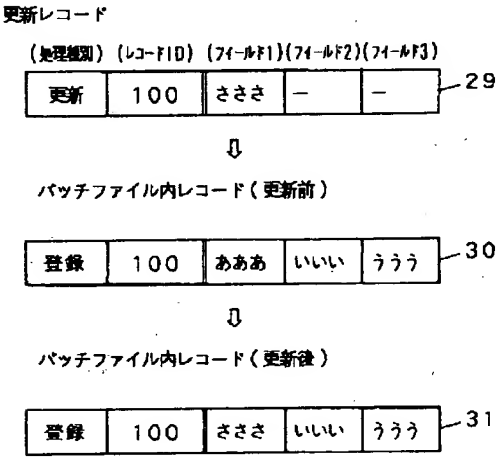
【図１】



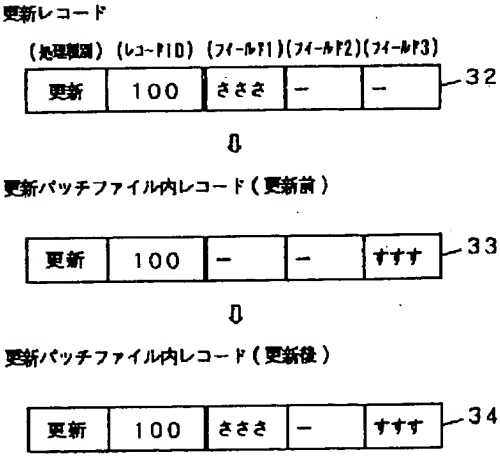
【図１５】



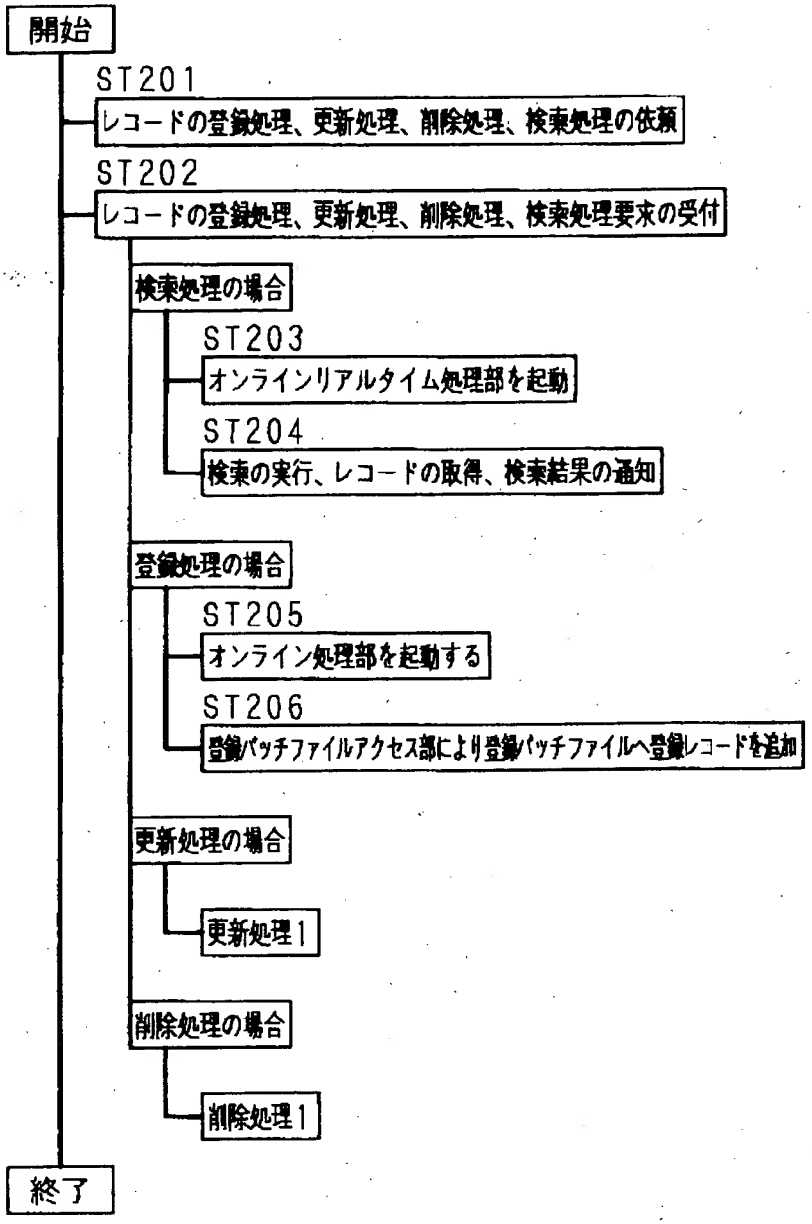
【図４】



【図５】



【図２】



【図８】

登録バッチファイル内レコード

(処理種別)	(レコードID)	(フィールド1)	(フィールド2)	(フィールド3)
登録	100	あああ	いいい	ううう
登録	101	かかか	きき き	くくく
登録	102	HHH	111	JJJ

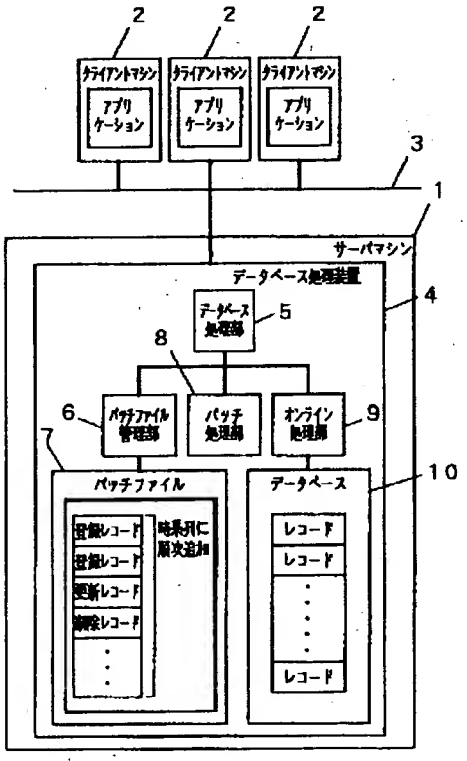
更新バッチファイル内レコード

レコードなし

削除バッチファイル内レコード

削除	1	−	−	−

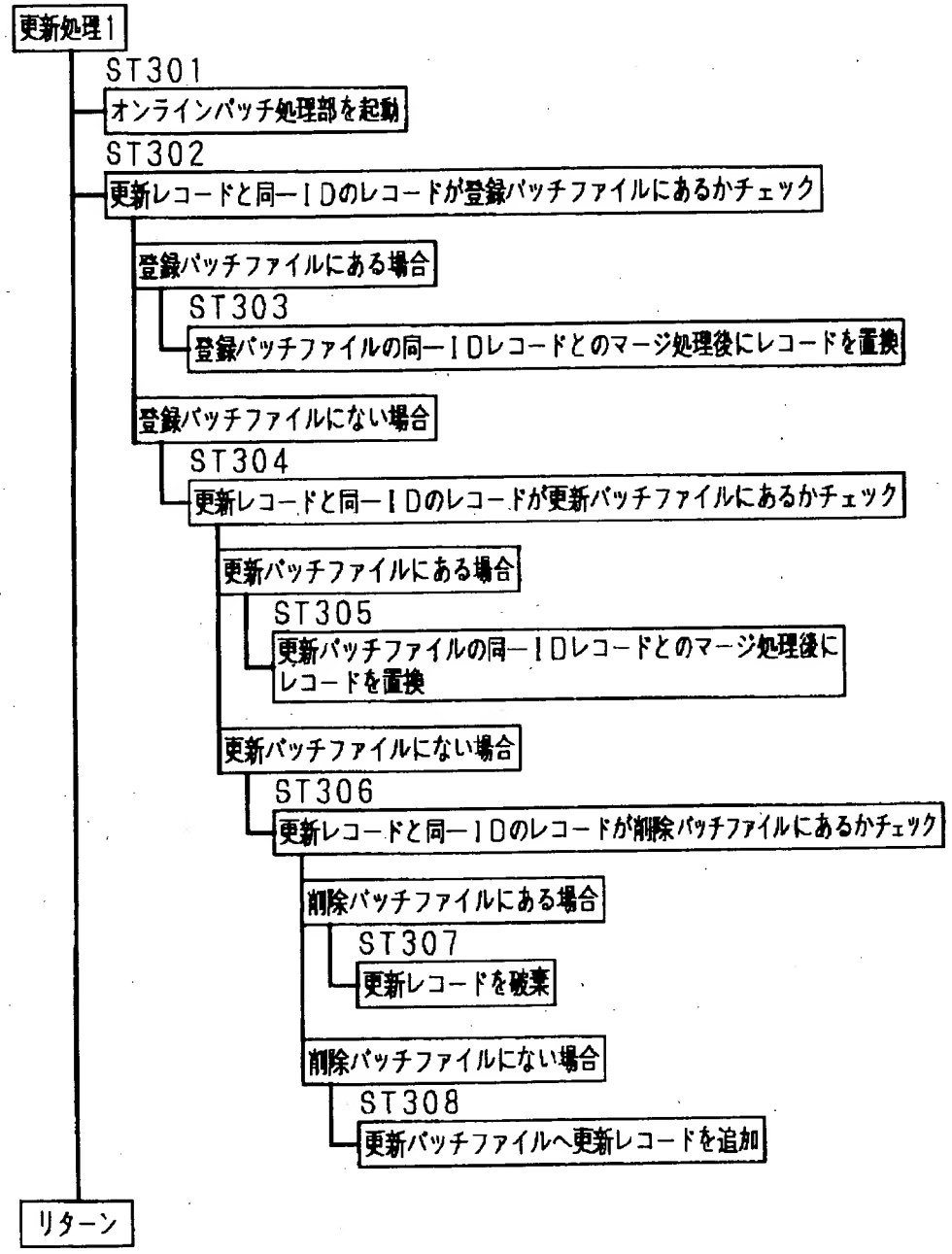
【図１４】



【図１２】

処理	規定時間	負荷なし時リアルタイム処理時間［対単位レコード長］	処理時間増加比率［対CPU負荷］	処理時間増加比率［対単位DBレコード数］
登録	7秒	1秒／100バイト	3%／1%	3%／1件
更新	8秒	0.5秒／100バイト	5%／1%	5%／1件
削除	12秒	1.5秒(固定)	7%／1%	7%／1件

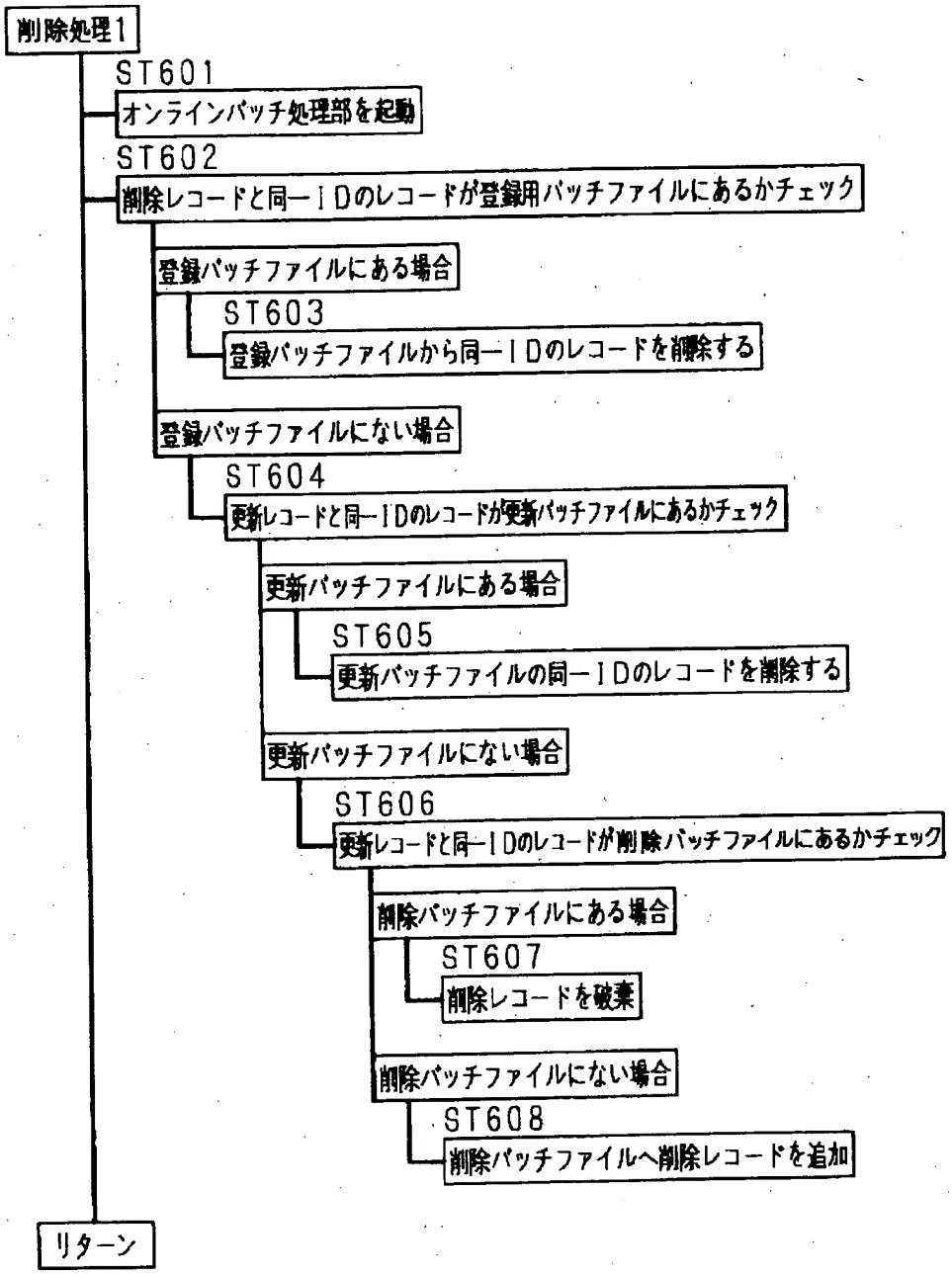
【図３】



【図１３】

$$\text{現負荷時リアルタイム処理時間} = \{\text{負荷なし時リアルタイム処理時間}[\text{対単位レコード長}] + (\text{負荷なし時リアルタイム処理時間}[\text{対単位レコード長}] * \text{CPU負荷状況} * \text{処理時間増加比率}[\text{対CPU負荷}]) + (\text{負荷なし時リアルタイム処理時間}[\text{対単位レコード長}] * \text{DBレコード件数} * \text{処理時間増加比率}[\text{対単位DBレコード数}])\} * (\text{レコード長}/\text{単位レコード長})$$

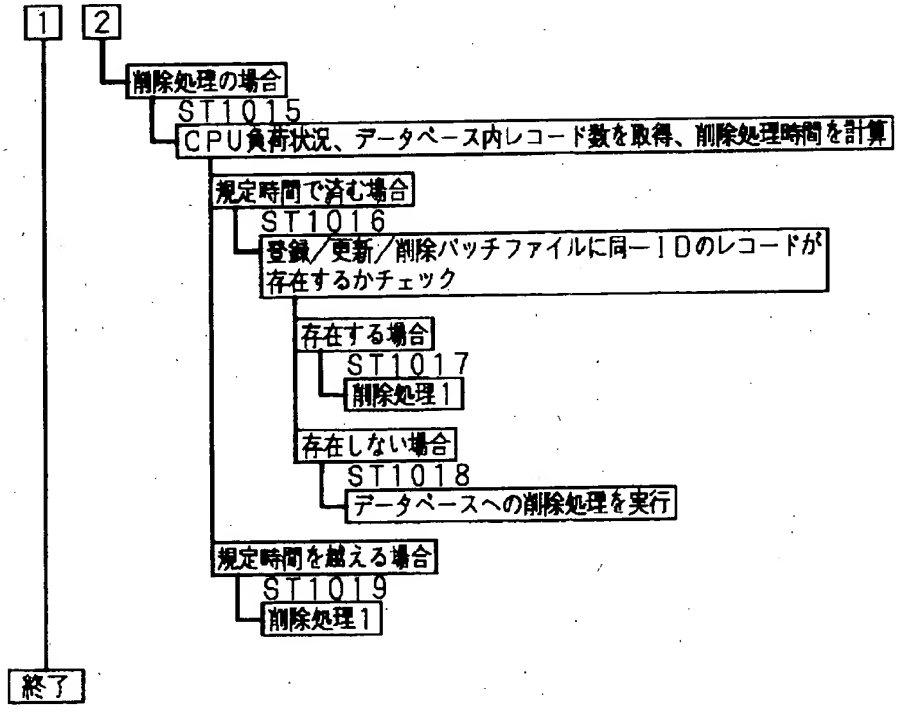
【図６】


削除処理１
ST601
オンラインバッチ処理部を起動
ST602
削除レコードと同一ＩＤのレコードが登録用バッチファイルにあるかチェック
登録バッチファイルにある場合
ST603
登録バッチファイルから同一ＩＤのレコードを削除する
登録バッチファイルにない場合
ST604
更新レコードと同一ＩＤのレコードが更新バッチファイルにあるかチェック
更新バッチファイルにある場合
ST605
更新バッチファイルの同一ＩＤのレコードを削除する
更新バッチファイルにない場合
ST606
更新レコードと同一ＩＤのレコードが削除バッチファイルにあるかチェック
削除バッチファイルにある場合
ST607
削除レコードを破棄
削除バッチファイルにない場合
ST608
削除バッチファイルへ削除レコードを追加
リターン

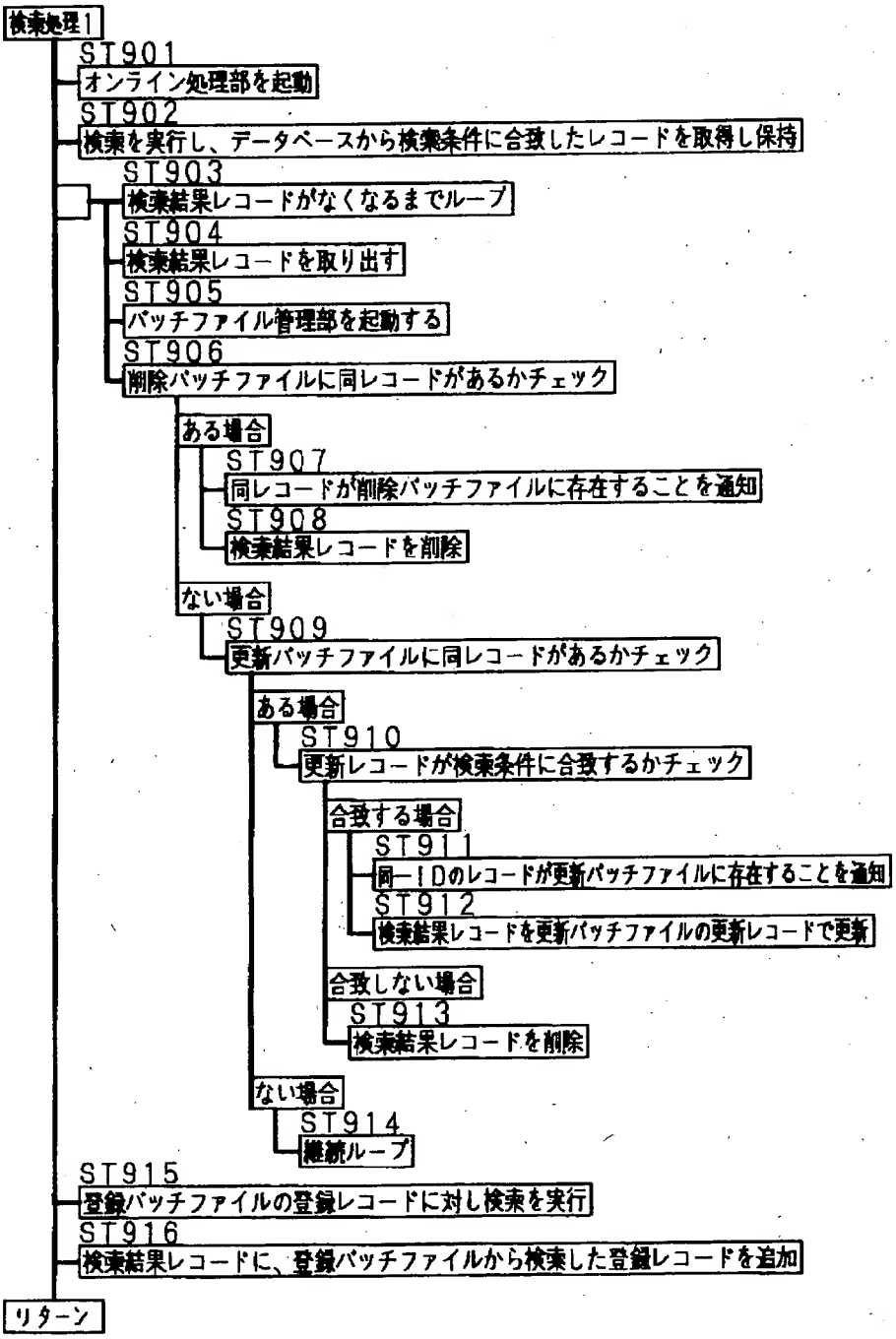
【図７】

(処理種別)	(レコードID)	(フィールド1)	(フィールド2)	(フィールド3)	
登録	100	あああ	いいい	うううう	……10:00
登録	101	かかか	きききき	くくくく	……10:10
更新	1	EEE	−	−	……10:20
更新	1	−	FFF	−	……10:30
更新	1	−	−	GGG	……10:40
更新	1	ささささ	−	−	……10:50
削除	1	−	−	−	……11:00
登録	102	HHH	III	JJJ	……11:10

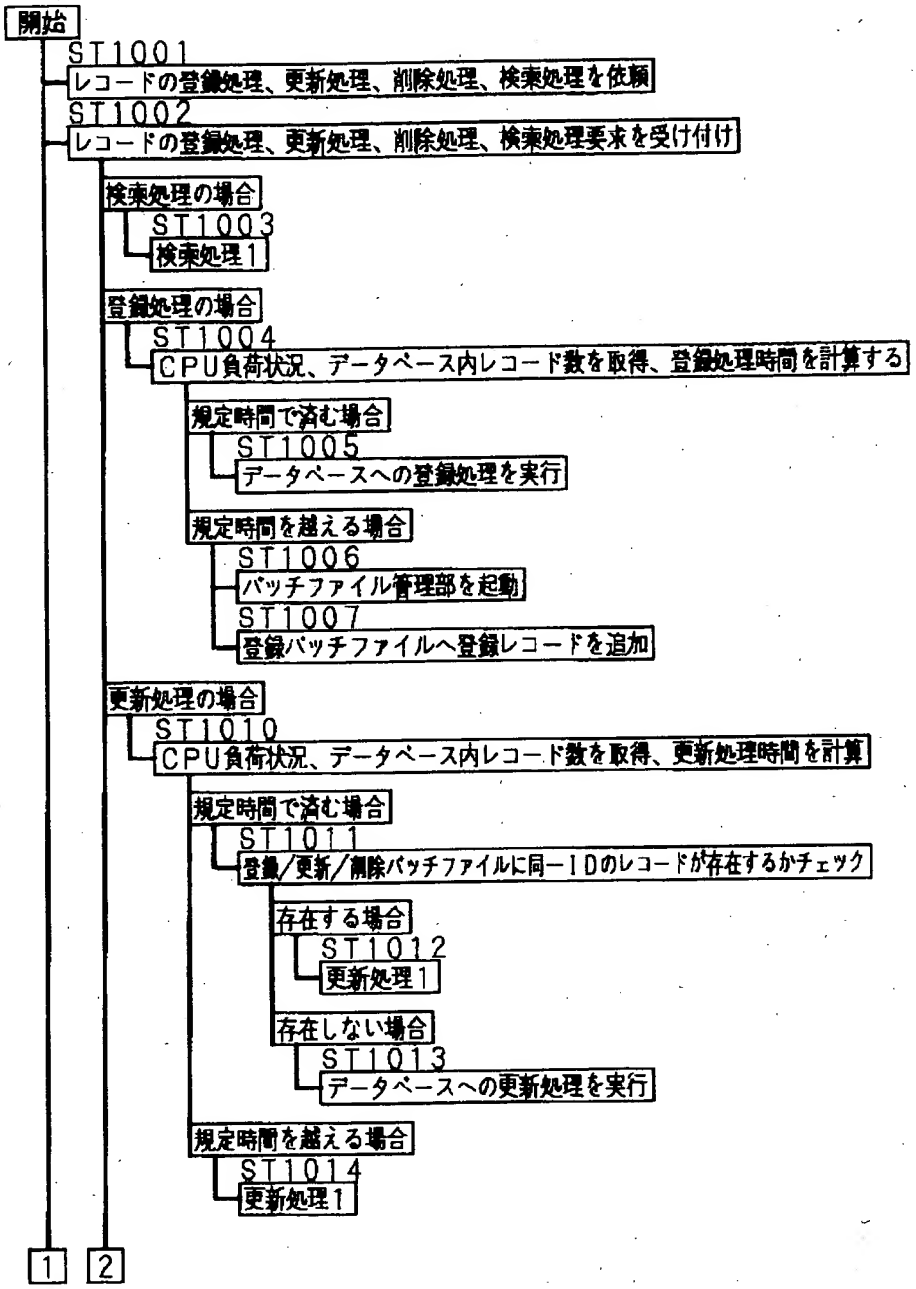
【図１１】



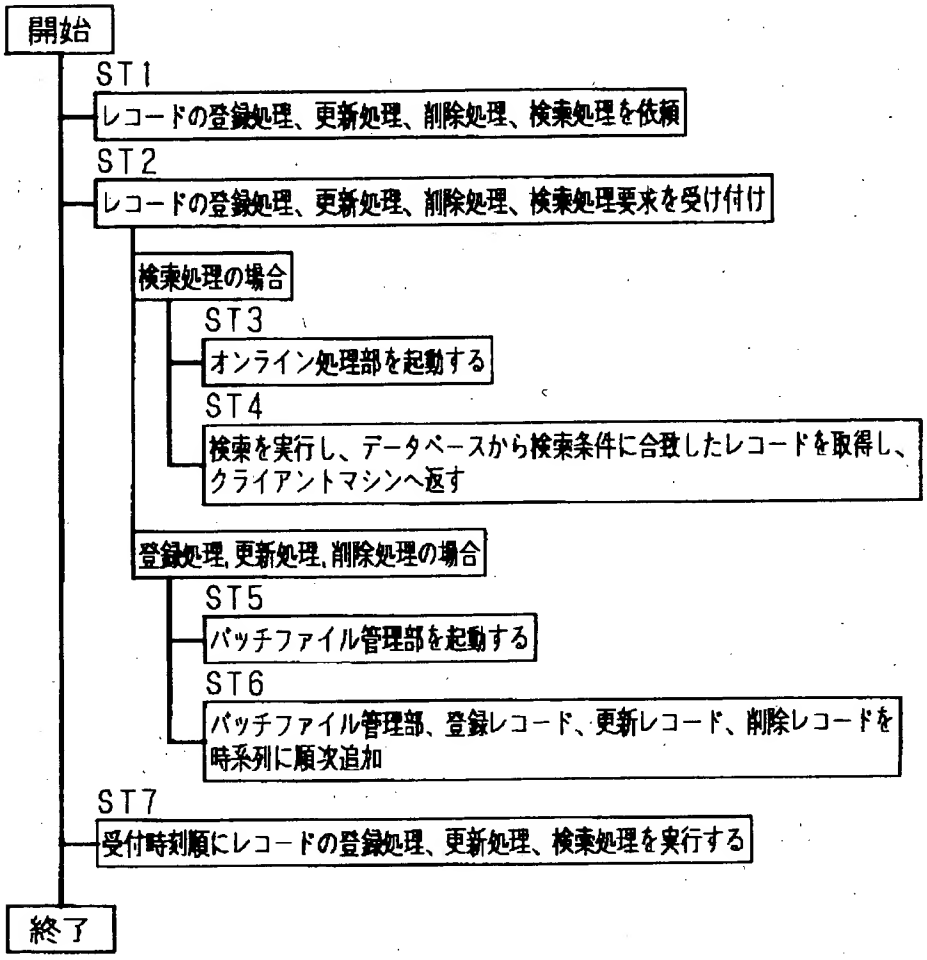
【図９】


検索処理1
ST901
オンライン処理部を起動
ST902
検索を実行し、データベースから検索条件に合致したレコードを取得し保持
ST903
検索結果レコードがなくなるまでループ
ST904
検索結果レコードを取り出す
ST905
パッチファイル管理部を起動する
ST906
削除パッチファイルに同レコードがあるかチェック
ある場合
ST907
同レコードが削除パッチファイルに存在することを通知
ST908
検索結果レコードを削除
ない場合
ST909
更新パッチファイルに同レコードがあるかチェック
ある場合
ST910
更新レコードが検索条件に合致するかチェック
合致する場合
ST911
同一IDのレコードが更新パッチファイルに存在することを通知
ST912
検索結果レコードを更新パッチファイルの更新レコードで更新
合致しない場合
ST913
検索結果レコードを削除
ない場合
ST914
継続ループ
ST915
登録パッチファイルの登録レコードに対し検索を実行
ST916
検索結果レコードに、登録パッチファイルから検索した登録レコードを追加
リターン

【図１０】


開始
ST1001
レコードの登録処理、更新処理、削除処理、検索処理を依頼
ST1002
レコードの登録処理、更新処理、削除処理、検索処理要求を受け付け
検索処理の場合
ST1003
検索処理1
登録処理の場合
ST1004
CPU負荷状況、データベース内レコード数を取得、登録処理時間を計算する
規定時間で済む場合
ST1005
データベースへの登録処理を実行
規定時間を越える場合
ST1006
バッチファイル管理部を起動
ST1007
登録バッチファイルへ登録レコードを追加
更新処理の場合
ST1010
CPU負荷状況、データベース内レコード数を取得、更新処理時間を計算
規定時間で済む場合
ST1011
登録/更新/削除バッチファイルに同一IDのレコードが存在するかチェック
存在する場合
ST1012
更新処理1
存在しない場合
ST1013
データベースへの更新処理を実行
規定時間を越える場合
ST1014
更新処理1
1
2

【図１６】



【図１７】

13 (処理識別)	11 (レコードID)	12 (フィールド1)	12 (フィールド2)	12 (フィールド3)	
登録	100	あああ	いいい	ううう	……10:00
登録	101	かかか	きき き	くくく	……10:10
更新	1	EEE	−	−	……10:20
更新	1	−	FFF	−	……10:30
更新	1	−	−	GGG	……10:40
削除	1	−	−	−	……10:50
更新	1	ささ さ	−	−	……11:00
登録	102	HHH	I I I	JJJ	……11:10